Durham Research Online

Deposited in DRO:

08 May 2009

Version of attached file:

Accepted Version

Peer-review status of attached file:

Peer-reviewed

Citation for published item:

Smith, Shamus P. and Trenholme, David (2009) 'Rapid prototyping a virtual fire drill environment using computer game technology.', Fire safety journal., 44 (4). pp. 559-569.

Further information on publisher's website:

https://doi.org/10.1016/j.firesaf.2008.11.004

Publisher's copyright statement:

Additional information:

Use policy

The full-text may be used and/or reproduced, and given to third parties in any format or medium, without prior permission or charge, for personal research or study, educational, or not-for-profit purposes provided that:

- a full bibliographic reference is made to the original source
- a link is made to the metadata record in DRO
- the full-text is not changed in any way

The full-text must not be sold in any format or medium without the formal permission of the copyright holders.

Please consult the full DRO policy for further details.

Durham University Library, Stockton Road, Durham DH1 3LY, United Kingdom
Tel : +44 (0)191 334 3042 | Fax : +44 (0)191 334 2971
https://dro.dur.ac.uk

Rapid Prototyping a Virtual Fire Drill Environment using Computer Game Technology*

Shamus P. Smith†, David Trenholme

Durham University, Durham DH1 3LE

United Kingdom

Abstract

Conducting fire evacuation drills in modern buildings under realistic fire conditions can be difficult. Typical fire drills do not feature dynamic events such as smoke filled corridors, fires in unexpected places or blocked fire exits that require on the spot decisions from evacuees. One alternative is the use of virtual environments. Virtual environments can support the training and observation of fire evacuee behaviours in 3D virtual buildings. However complex virtual environments can be difficult to build. This paper explores how the reuse of computer game technology can aid in the rapid prototyping of virtual environments which can be populated with fire drill evacuation scenarios. Over a three week period, a single developer constructed a realistic model of a real world building to support virtual fire drill evaluations. While participants in a user study found the simulated environment realistic, performance metrics indicated clustering in the results based on participants previous gaming experience.

Keywords: Virtual reality; Virtual fire drill; Fire evacuation; Game engines; Rapid prototyping; Evaluation.

1 Introduction

Conducting fire evacuation drills in modern buildings under realistic fire conditions can be difficult [11]. During fire drills, building occupants typically exit at a leisurely pace, without having to deal with panic inducing events such as smoke filled corridors, fires in unexpected places or blocked fire exits. Ko et al. [15] observe that there "is always an uncertainty regarding the exact situation evacuees would find themselves in an emergency evacuation". In addition, fire evacuation drills are either considerably

*Fire Safety Journal, Volume 44, Issue 4, May 2009, Pages 559-569, doi:10.1016/j.firesaf.2008.11.004

†Contact author - shamus.smith@durham.ac.uk, tel. +44(0)191 334 4284, fax +44(0)191 334 1701

disruptive when they occur unannounced or tediously uneventful if routinely practiced [30]. There has been considerable work done on modelling human behaviour in response to fire events (for recent examples see [10, 20, 24, 26, 32, 45]) but detailed evacuation behaviour of individual building occupants can be somewhat unpredictable [27, 30].

One alternative is the use of virtual environments to simulate fire evacuation scenarios. However building realistic virtual environments is a complex, expensive and time consuming process [17, 31, 37]. In addition to generating virtual object models [14], for example buildings and scenery, a developer must also manage support for user interaction [5], any dynamic or world behaviours that are required, for example collision detection, and any non-visual features of the virtual world, for example audio cues [25]. Although virtual environment development toolkits are available, many only provide a subset of the tools needed to build complete virtual worlds [4]. Some features of virtual worlds such as wind, fire, smoke and water, and the provision for embodied autonomous agents [1], are particularly hard to simulate. In addition, virtual environment toolkits often require additional programming skills and a substantial time investment on the part of the developer.

In recent years, work in the video game and virtual reality industries has been overlapping [2, 47]. The current generation of computer games present realistic virtual worlds featuring user friendly interaction and the simulation of real world phenomena, for example gravity. Using computer games as the basis for virtual environment development has a number of advantages. Computer games are robust and extensively tested [18], both for usability and performance, work on off-the-shelf systems [31] and can be easily disseminated, for example via online communities. Many computer game developers support modification of their game environments by releasing level editors, i.e. to modify the game environment, and tools to edit the game behaviour. This allows the reuse of the underlying game engine technology, including 3D rendering, 2D drawing, sound, user input and world physics/dynamics [19]. These advanced features allow game engines to create realistic simulations and the modular nature of game engines means they are general enough to be adapted for different games or applications, which allow them to be reused [19].

This paper describes the use of commercial off-the-shelf computer game development tools to rapidly develop a virtual environment to test fire drill behaviours in a 3D virtual model of a real world building. We explored how reusing computer game technology can simplify the modelling of virtual buildings, the generation of effects such as fire and smoke - which are typically difficult to program - and the inclusion of realistic audio cues, such as fire alarms. A user study was conducted in three fire drill scenarios to evaluate the usability and realism of the developed virtual environment.

The remainder of this paper is as follows. Section 2 describes related work and Section 3 defines a number of computer game technologies that were considered. Section 4 describes the development of the virtual fire drill environment and Section 5 outlines the evaluation study. The results of the evaluation study are described in Section 6, followed by a discussion in Section 7. Conclusions and future work are presented in Section 8.

2 Related Work

There are three areas of related work relevant to the research presented here; model-based fire evacuation tools, virtual reality for fire evacuation simulation and the reuse of computer game technology.

2.1 Model-based Fire Evacuation Tools

Fires in buildings and enclosed areas are a high consequence hazard that can result in the loss of life. Therefore there is continuing interest in the use of computer simulations to identify preventative measures for fire risks. This includes simulations and tools to support (i) human behaviour modelling in the presence of fires and other dangers such a terrorist threats, (ii) the design and operation of structures in the presence of fires, and (iii) modelling the spread of fire and smoke[1].

It is common for mathematical models to be used to simulate the evacuation of large numbers of individuals from enclosed spaces. One example of this is the EXODUS system [8]. EXODUS is based on an expert system that predicts the motion and behaviour of individuals with a set of heuristic rules. The system uses the interaction of several subsystems, namely movement, behaviour, passenger, hazard and toxicity, to predict evacuation results. Recent updates to the system have included the capability to model the impact of smoke, heat and toxic gases on the physical abilities of evacuees and to allow the modelling of adaptive behaviour from the simulated evacuees, for example modified behaviour based on an evacuees' familiarity with the building they are in [10].

Shen [32] considers the Evacuation Simulation Model (ESM), which is a network model consisting of stocks - representing rooms, corridors etc. - and flows - passages between objects represented by stocks. ESM can simulate the optimised evacuation, the phased evacuation and the evacuability of a building, as the percentage of successful evacuees to the total occupants. Therefore, ESM can evaluate the evacuation safety of buildings. ESM is a useful tool for evaluating evacuation performance and the safety of a building because of its flexibility and other features, such as tracing the movement of occupants and predicting possible victims. Also see [44] where simulation efficiency is considered in modelling buildings with a fine grain representation of zones of interest, for example the fire floor, and

[1]A full summary of computer simulations for building evacuation is outside the scope of the current paper and the reader is directed to [11, 44].

a more course grain representation for the remain parts of the building.

Johnson [12] describes the use of the Glasgow Evacuation Simulator (GES) which was specifically developed to model the behaviour of large groups of people. In [12] the GES is used in the context of terrorist threats which has many commonalities to fire evacuation modelling, for example from modelling the evacuation of spectators from an indoor event to modelling security staff as an analogy to fire rescue teams. Johnson and Nilsen-Nygaard [13] consider evacuation simulators in the context of counter terrorism, and in particularly in modelling of human behaviour in response to improvised explosive devices. An important benefit of such tools is that they can be used to explore alternate intervention strategies as part of the training and planning of emergency service teams [13].

Metro station fire safety analysis in China is considered in [46]. The simulated analysis of occupant evacuation from metro stations can provide a reference for both the design and safe operation of metro stations. Through simulations, a benchmark for the evacuation of deeply buried metro stations has been developed to show minimum requirements for occupant safety based on predictions of the quality of smoke produced.

Qin et al. [28] consider how numerical simulations play an important role in investigating the spread of smoke in large buildings. They note that smoke is often the main cause of deaths in an indoor fire. In the context of a fire in an atrium, [28] describes different natural and mechanical smoke exhaust systems.

Fire simulation can also be an important tool for fire investigation. Shen et al. [33] observe how fire simulation tools can "reconstruct fire processes, explain fire development and demonstrate smoke movement through describing the configuration of fuel, effects of ventilation, design of the building, the impact of manual or automatic systems and fire source." Fire scene reconstruction supported by computer simulation can provide information for fire investigators in a way that is not easily supported through traditional fire scene investigation techniques.

The majority of fire evacuation simulators are used to predict evacuee behaviour or to critique the layout of buildings or other environments that require emergency evacuation. In order to argue for realism in the simulations, the rules governing the simulated evacuees' behaviour are typically compared to other models [20, 44], to government evacuation requirements [46], to data of actual evacuations [8, 10] or through comparison to known evacuee behaviours [13, 24, 26, 45]. In contrast, the work here considers fire evacuation in the context of simulated fire drills. We are interested in evaluating the evacuee behaviour of *actual humans* to simulated fire threats. However, conducting fire evacuation drills in modern buildings under realistic fire conditions can be difficult. One alternative is the use of virtual reality simulations with actual human participants.

2.2 Virtual Reality for Fire Evacuation Simulation

Ren et al. [30, 29] present a virtual environment developed to simulate the evacuation of an underground station. The article highlights two main reasons why evacuation may fail: poor layout of the building or improper action taken by occupants due to panic or unfamiliarity with evacuation procedures. Virtual environment training can be used to help identify problems with layout and help familiarise occupants with evacuation routines, which would be dangerous and expensive in a real building. The virtual environment was developed in Vega[2], a real-time scene simulation platform. However, Ren et al. [30] observe that Vega did not provide a complete set of the features needed to build virtual worlds and separate software was needed to model the 3D building, in Multigen Creator[3], and generating fire/smoke effects, in the Fire Dynamics Simulator[4].

Louka and Balducelli [21] describe the use of virtual reality technology in the context of emergency operation support and training as the only feasible alternative to full-scale fire tests inside tunnels. Data relating to the simulation can be managed by a single database and other simulation tools, e.g. for fire and smoke, can act on this data. Louka and Balducelli [21] discuss the different types of hardware and software available and how they can affect presence [36], the degree to which the user accepts the computer generated experience as real, and immersion, which depends on the hardware being used. A distinction is made between virtual reality training simulators - where the knowledge encoded in such systems is only used to present a realistic environment to move around in - and training systems - where training functionality is integrated, which guides the user to teach a correct procedure or remedy incorrect actions. By this classification, the system considered in the work described here is a virtual reality training simulator.

A virtual reality based feasibility study of evacuation time is compared to a traditional, mathematical, method for calculating evacuation planning in [34]. A virtual reality model of a building is created, using 3D WebMaster, to compare time/reactions of a user to that calculated mathematically, in a number of different scenarios, e.g. with and without smoke/signs. The study concluded that a virtual reality simulation is a viable alternative to traditional calculation methods in evaluating zoning plans, evacuation times and location of exits. In addition to showing that virtual environments are useful in this kind of application, Shih et al. [34] demonstrate that a virtual environment can help provide results that might not otherwise be predicted by mathematical calculations.

Tang et al. [41] describe the use of a virtual environment to investigate the influence of various emergency signs in aiding way-finding behaviours of building users in the event of an emergency. Three

[2] http://www.multigen.com/products/runtime/vega_prime/ [last access 12/11/2008].
[3] http://www.multigen.com/products/database/creator/index.shtml [last access 12/11/2008].
[4] http://www.fire.nist.gov/fds/ [last access 12/11/2008].

scenarios were tested with old-version emergency signs, new-version emergency signs and a control group with no emergency signs. The groups without signs took far longer to exit the virtual building and there were gender differences in way-finding ability with males exhibiting better way-finding skills. There is little information in the paper about the virtual environment apart from that it was developed in Quest 3D[5], a commercial tool for creating real-time 3D applications.

In contrast to actual fire drills, virtual environment overcome the disadvantages of high cost, poor repetitiveness and danger [29]. However the development of realistic virtual environments themselves can be a complex, expensive and time consuming process. In addition, developing virtual environments typically requires a number of specialist developer skills and access to, and the integration of, multiple sub components. For example the system described in [29] requires access and knowledge of (i) Vega, a real-time scene simulation platform, (ii) Multigen Creator, a 3D modeller used for the virtual building, (iii) Fire Dynamics Simulator, used to provide fire science support, and (iv) Visual C++.Net, for custom code development.

The work in this paper investigates an alternative approach in the reuse of computer game assets and the use of existing off-the-shelf game engines to rapidly develop and support fire drill simulations. This follows recent trends in the reuse of computer game technology in non-game applications.

2.3 Computer Game Technology Reuse

Mac Namee et al. [22] describe the development of *Serious Gordon*, a virtual environment to teach food safety. The Source engine developed by Valve Corporation[6], used in the game Half-Life2[7], was reused in the construction of the environment. Mac Namee et al. [22] discuss how particular attention was paid to the story of the game to provide an interesting scenario that was also realistic and covered the necessary learning outcomes. Also they provide an overview of how game play elements from the base game, Half-Life2, that were deemed unnecessary were removed and other game assets were modified. The major challenge identified was the need to balance the competing goals to teach and to entertain.

A modification of Quake III Arena[8] was used to test the suitability of using computer games for psychological experimenting. Frey et al. [7] found that using virtual environments to collect data can compromise the internal validity of the data as some people have previous experience with virtual environments, e.g. people who play computer games. An investigation was carried out to see if training in a virtual environment can help reduce these individual differences in user performance and through the modification of Quake III Arena, several virtual environments were built. Frey et al. [7] found that,

[5] http://www.quest3d.com/ [last access 12/11/2008].
[6] http://www.valvesoftware.com [last access 12/11/2008].
[7] http://www.Half-Life2.com/ [last access 12/11/2008].
[8] http://www.idsoftware.com/games/quake/quake3-arena/ [last access 12/11/2008].

after training, the performance of non-players and inexperienced players had improved to a greater extent than that of experienced players which partially removed the differences in performance due to previous experience.

The use of game engines in producing a visual walkthrough application for a virtual office is described in [35]. The virtual office was designed in a 2D CAD application before being converted to 3D and imported into UnrealED, the map/level editor for the Unreal Engine game engine[9]. Shiratuddin and Thabet [35] note several advantages of using game engines for creating virtual walkthroughs, such as low cost, network support, collision detection, support for a high frame rate and relatively low hardware requirements. Also they note that the use of a 3D game engine can greatly improve real-time walkthrough experiences as needed in virtual reality applications.

Bell and Fogler [3] describe an ongoing project to create a series of virtual reality based simulations for laboratory accidents using a variety of different platforms. The project is based on the idea that real life experiences of lab safety procedures and accidents will have a greater impact than just reading rules and guidelines. Since it is too expensive and dangerous to simulate this in real life, a virtual reality simulation is used to be as close as possible to real life experiences. One of the scenarios, the proper storage of chemicals, has been implemented using the Half-Life[10] game engine. The Half-Life game engine was suitable in this case because it is easily modified, provides sophisticated lighting effects, can trigger actions by user movement and show explosions.

3 Computer Game Technology

A number of computer game developers provide tools, documentation and source code, either with the game itself or separately available, so that end-users can create new content for the game called a *mod* or *modification* [9]. For example, users can create new levels, maps, items or characters and add them into the game, known as a *partial conversion*, or create an entirely new game by altering the games source code, know as a *total conversion*.

This section overviews several currently available game engines that are suitable for prototyping virtual environments (also see [42]). Modern game engines have a modular structure so that they can be reused for different games [19]. Computer games in the First Person Shooter (FPS) genre tend to have the greatest capability and resources for modification, so this section will examine game engines for this game genre. There is a wide selection of 3D game engines available for potential reuse[11]. Lewis and Jacobson [19] observe that there are more than 600 commercial game engines. However, the game

[9] http://www.unrealtechnology.com/ [last access 12/11/2008].
[10] see http://www.valvesoftware.com/games.html [last access 12/11/2008].
[11] For a comprehensive database of 3D engines see http://www.devmaster.net/engines/ [last access 12/11/2008].

engines considered here are those used in the most recently available commercial computer games, thus representing more advanced technology which usually results in more realistic environments.

3.1 CryENGINE

The *CryENGINE* was created by software developers Crytek[12] and used in the game Far Cry[13] released in 2004. The CryENGINE supports a number of features that are useful for creating immersive and realistic games and virtual environments, such as a real-time editor, dynamic lights, a network system, an integrated physics system, shaders, shadow support and a dynamic music system. The necessary development tools are integrated with the engine itself, including the *CryENGINE Sandbox* world editing system. Licensed developers receive full source code and documentation for the engine and tools[14].

Advantages to using the CryENGINE include that the engine produces very high quality graphics and visuals. The necessary development tools are provided with the engine and so can be accessed from games that use the engine, such as Far Cry. Also, the Sandbox editor is a very intuitive tool as it edits levels in real time, offering *what you see is what you play* feedback. Partial source code and documentation is included with a freely downloadable SDK[15]. A disadvantage with this engine is that it has high demands on supporting hardware requirements for the high quality visual and audio components.

3.2 id Tech 3 Engine

The *id Tech 3 engine* is a game engine developed by id Software[16] and was first used in the game Quake III Arena[17], released in 1999. Designed to be the ultimate multiplayer experience, Quake III Arena has become a defacto standard for professional gamers and is the common choice for gaming tournaments around the world.

Advantages to using this engine include that in addition to the Quake III game source code, the engine source code has been released under the GNU General Public License (GPL). The availability of the full source code provides more flexibility in the customisation of game environments. Additionally there are a large number of tutorials and articles that have been written for this engine and with the release of the source code in 2005, there is still an active development community. One disadvantage of using this engine is that game engine technology has progressed greatly since its original release in

[12]http://www.crytek.com [last access 12/11/2008].
[13]http://www.farcry-thegame.com [last access 12/11/2008].
[14]http://www.crytek.com/technology/cryengine-2/specifications/ [last access 12/11/2008].
[15]http://crymod.com/filebase.php [last access 12/11/2008].
[16]http://www.idsoftware.com [last access 12/11/2008].
[17]http://www.idsoftware.com/games/quake/quake3-arena/ [last access 12/11/2008].

1999, and more advanced engines are available for use [16].

3.3 id Tech 4 Engine

Developed by id Software, the *id Tech 4 engine* was first used in the game Doom 3[18]. id Tech 4 was previously known as the Doom 3 engine. The id Tech 4 engine provides several unique features, such as a unified lighting paradigm, where every surface uses the same rendering pipeline, leading to visual consistency and fully dynamic shadows. The engine also makes short map compilation times possible. The level editor for this engine is built into games that use it and so can be accessed by anyone who owns any of these games.

The main advantage of using the id Tech 4 engine is that it is a very powerful engine, capable of producing realistic environments. Also, there is a large modding community, including a large number of tutorials and articles are available. However, similar to CryENGINE (Section 3.1), the hardware requirements of the engine are high, for example requirements for Doom 3 include: 100% DirectX 9.0b compatible 64MB hardware accelerated video card, Pentium IV 1.5 GHz or Athlon XP 1500+ processor or higher, 384MB RAM, 2.2GB of uncompressed free hard disk space (plus 400MB for Windows swap file) and 100% DirectX 9.0b compatible 16-bit sound card.

3.4 Jupiter Extended (EX)

Jupiter EX is the latest version of a multi-platform game engine and development kit developed by Touchdown Entertainment[19] and used for the game F.E.A.R[20]. Jupiter EX includes a set of content creation tools, each with a different purpose. *WorldEdit* allows designers to create 3D environments and add objects and lighting. *ModelEdit* allows the designer to optimise models so that they interact correctly with the environment and other objects. *FxEdit* can create and modify visual effects, without requiring re-compilation of the game, which increases the efficiency of the implement/test cycle for special effects. Other tools include *ArchiveEdit*, a tool used to package game assets into one archive for distribution and *GDBEdit*, the game database editor for changing things such as weapon damage, the interface etc. without changing source code.

The advantages of modifying games that use the Jupiter EX engine include the high level of graphical fidelity that can be achieved, which will improve realism and immersion. Also, support tools required for modifying Jupiter EX games are available and logically separated into a number of different applications and come with documentation. The main disadvantage is that the modding

[18] http://www.idsoftware.com/games/doom/doom3/ [last access 12/11/2008].
[19] http://www.touchdownentertainment.com/ [last access 12/11/2008].
[20] http://www.whatisfear.com [last access 12/11/2008].

community is not as large for F.E.A.R, the most popular Jupiter EX game, as it is for other games such as Half-Life2 or Unreal Tournament 2003/2004.

3.5 Source Engine

The *Source Engine* was developed by the Valve Corporation[21] and is most notably used in games such as Half-Life2[22] and Counter-Strike: Source[23]. The Source Engine features a high degree of modularity and flexibility, lip synchronisation and facial expression technology and a realistic physics system. The engine also supports particle effects, volumetric smoke and environmental effects such as fog and rain. An advanced artificial intelligence system is provided which enables sophisticated character navigation, enabling characters to run, jump, climb stairs etc.

The level editor that is used to produce levels for games using the Source Engine is the *Valve Hammer Editor*, commonly referred to as *Hammer*. All the necessary tools for creating a mod and game content, e.g. via Hammer, are included with the Source SDK, which is available after purchasing a Source-based game, e.g. Half-Life2, and can be downloaded using Valve's digital content delivery service called *Steam*[24]. Documentation for the Source SDK is provided on the Valve Developer Community[25] wiki, which was created by Valve and is the definitive and most comprehensive source of information on using the Source Engine.

3.6 Unreal Engine 2

A powerful and widely used game engine developed by Epic Games[26], *Unreal Engine 2* was used to develop Unreal Tournament 2003[27] and its successor, Unreal Tournament 2004. *UnrealEd 3*, the level editor, is included with Unreal Tournament 2003/2004. The engine supports high performance rendering, advanced animation features and high-quality dynamic lighting.

Modification of games based on the Unreal Engine 2 is further supported by inclusion of sample content and the source code for the engine and editor. The engine source code is written in C++. The engine also includes a scripting language, *UnrealScript*, which can modify various aspects of the gameplay[28]. Many resources are available for modifying Unreal Engine based games, including Epic's *Unreal Developer Network (UDN)*[29], which is the official support site for licensees and mod developers and provides technical documentation as well as tutorials for the Unreal Engine and UnrealEd.

[21] http://www.valvesoftware.com [last access 12/11/2008].
[22] http://www.Half-Life2.com/ [last access 12/11/2008].
[23] http://counter-strike.net/ [last access 12/11/2008]
[24] http://www.steampowered.com [last access 12/11/2008].
[25] http://developer.valvesoftware.com/ [last access 12/11/2008].
[26] http://www.epicgames.com/ [last access 12/11/2008].
[27] http://www.unrealtournament3.com/ [last access 12/11/2008].
[28] see http://www.unrealtechnology.com/ [last access [12/11/2008].
[29] http://udn.epicgames.com [last access 12/11/2008].

3.7 Game Engine Selection

All the engines considered in this section are suitable for developing non-game virtual environments. Lewis and Jacobson [19] note that in many cases it would be hard to imagine an application for which one game engine would be suited and others not. Particle effects, which are needed to simulate fire and smoke in the target virtual environment, are supported in all the engines considered. Also all the engines provide some source code to users; usually partial source code that can be altered and recompiled to affect the actual game play.

The Source engine was chosen for the fire evacuation simulator as it provides everything required to produce a high quality virtual environment and has extensive support included with the Source software developers kit (SDK), e.g. the world editor Hammer and the games source code. The technical features of the Source engine were also more than adequate for the requirements of the problem domain, e.g. support for particle effects such as fire and smoke and additional properties that are particularly relevant to a fire drill simulator including allowing fire entities to ignite other fire entities, giving the impression of a spreading fire, which will help increase the realism of the environment.

4 Virtual Fire Drill Environment

4.1 Building Layout

The building chosen to be modelled was the Computer Science department at Durham University. The department building consists of three floors with a number of offices, meeting rooms and a reception on the first floor. Three exits were identified: the main entrance on the ground floor, a rear exit on the first floor and a connecting corridor between Computer Science and the Engineering department on the second floor.

Floor plans of all three floors of the department were used to determine the layout and relative scale of corridors, doors and rooms. Elements in the real building that would act as audio and visual cues to someone during a fire evacuation were identified in the department and their positions added to the plans so that they could be accurately included in the final model. These included fire alarm triggers, fire alarm sirens, fire exit signs and fire extinguishers. Photographs of various areas of the department were taken to aid texturing the 3D model.

4.2 Fire Simulation

One of the main features required of the game engine was support for fire and smoke effects. Using the Source Engine and Hammer, this was accomplished by using *fire entities*, which are objects that generate fire at their origin and smoke effects above the flame. The smoke can be toggled on/off, the height of the flame can be specified and fires can be set to have an infinite duration, i.e. fires that will burn forever. Figure 1 shows views of the real building and the virtual building with a collection of fire entities positioned close together.

Figure 1: Virtual model view (left) including fire entities and accompanying smoke and real world view (right) of a stairwell.

The spread of fire through the virtual environment is an important part of the fire evacuation simulations. This was achieved by placing several fire entities in close proximity to each other and only igniting the fire entities at the fire start point. Although this did not produce a completely accurate representation of how a fire might spread in a real building, it was a useful method for quickly producing a fire that spreads over time.

4.3 3D Virtual Building Model

The main geometry for the model of the Computer Science department was created using Hammer, the level editor for games using the Source engine (see Figure 2). The three main floors of the department were created separately, using photographs and floor plans for reference. Most of the offices themselves were not modelled, on the assumption that their doors would be locked and they would be unreachable in the final model.

Once the level geometry was completed, basic object props were added to the model to make it more accurately represent the department and increase its realism. Lights and light fittings were added to the ceiling, chairs and other furniture, such as filing cabinets, corkboards and shelves, were placed in appropriate places in the corridors (see Figure 3). Glass windows were added in some areas, such as the main office reception.

Source engine games, such as Half-Life2, provide a number of resources that can be reused in their modifications. However there was a problem with the story genre of the game being modified. Since Half-Life2 is set in a futuristic, dystopian world, many of the props available to user modifications reflect this, i.e. they appear damaged or “distressed” to some extent, and would look out of place in a

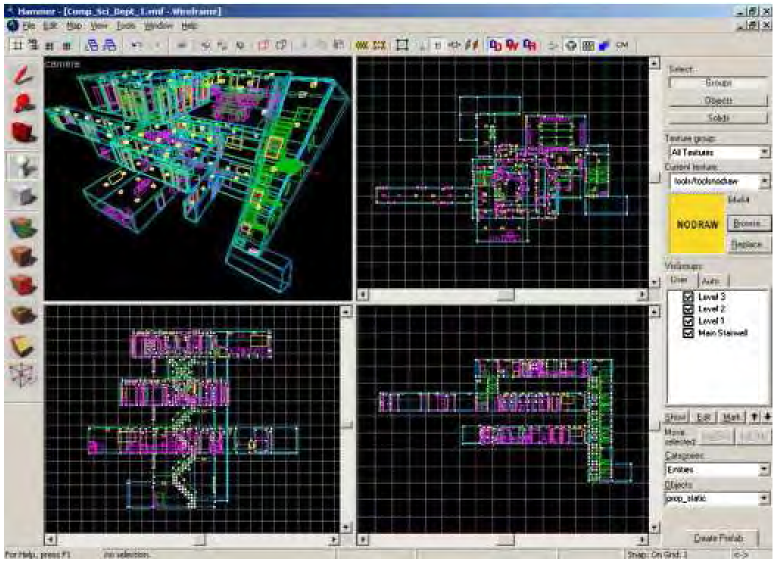

Figure 2: The completed model geometry in the Hammer editor.

Computer Science department. In some cases, this limited the assets that could be reused from the original game and meant that new, customised media had to be created.

Figure 3: Virtual model (left) and real building (right) views of the Computer Science Department.

4.4 Fire Drill Evacuation Scenarios

Three fire drill evacuation scenarios were created. This was achieved by having three copies of the same map, so that the 3D building model was identical in all three scenarios, but with fire entities placed in different locations to define unique escape routes for each scenario. In each scenario, a trigger is placed in the staffroom - on the top floor of the building - so that when the user enters, a timer will start and the user is told to wait. Three seconds later, a fire breaks out triggering an audio cue fire alarm and closing all the smoke stop doors in the corridors, which are initially open.

When the fire alarm sounds, the user is instructed to find an exit, and in doing so must navigate through the virtual building avoiding any routes blocked by fire and smoke. There is only one viable exit in each scenario, which is only enabled once the fire starts. When the user reaches an exit, another trigger causes the virtual environment to fade out and the user is informed of their success and the scenario ends.

Explicit fire science, as found in fire evacuation simulators such as [10, 28, 33], is lacking from the three scenarios. Due to the exploratory approach to the prototype development only limited fire and smoke, as provided by the base game technology, was included. Smoke is explicitly linked to fire entities so that where there are more fire entities there is thicker smoke. Also the fire entities were set to expand as time progressed. Therefore users in the environment who returned to a fire site would find both a larger fire and more smoke[30].

The full 3D model and fire drill scenarios were completed in three weeks by a single developer (the second author). This is indicative of the support provided by the game technology and the rapid nature of reusing game assets. This allowed the developer to concentrate on the content construction and the prototyping of the evacuation scenarios.

In all three scenarios, the user, navigating the building in first-person perspective, is the only person in the virtual world. When evaluating the developed environment we were primarily interested in usability and environment realism. Early trials of the environment featured computer controlled non-player characters (NPCs). However, initial experiments with NPC movement, based on game supported scripting, had NPCs being trapped in rooms with fires. It was felt that this would distress users in an evaluation study and interfere with their own evacuation from the virtual environment. Therefore no NPCs were included in the user study environments[31].

A user study of the fire drill evacuation scenarios was conducted to investigate the usability and realism of the constructed virtual environment and whether the computer game development tools are suitable for rapidly prototyping virtual environments. The study aimed to collect performance metrics of building evacuation times, subjective responses from questionnaires and video/audio data for later analysis.

5 Evaluation Study

Twelve participants volunteered for the study and were recruited through university mailing lists. They consisted of 2 females and 10 males with ages ranging between 18 and 52. All participants used computers on a daily basis and for work activities. Eleven participants used computers for educational activities and nine participants used computers for entertainment. Seven participants reported

[30]The lack of fire science provided by the game technology will be discussed further in Section 7.

[31]The use of NPCs and the inclusion of other human participants is part of our future work and will be considered in Section 8.

frequent, i.e. at least once a week, computer game usage describing themselves as novices (0), experienced players (5) and expert players (2). Participants were paid four pounds (UK£) for participation and all twelve participants completed the three fire evacuation scenarios. The evaluation study was approved after independent scrutiny by the Ethics Committee of the Department of Computer Science, Durham University.

The evaluation was carried out in an empty office. The equipment consisted of an ERGO Notebook PC with Twin Intel 1.66GHz CPU's, 504MB of RAM, Windows XP SP2 and a standard mouse. The fire evacuation mod was loaded into the single player version of Half-Life2 in offline mode. Navigation in the environment was by the arrow keys on the Notebook PC and a standard mouse.

5.1 Procedure

Demographic, computer usage and computer game usage data was gathered in a pre-session questionnaire. A consent form was also signed by all participants. This included explicit permission for the use of video footage collected during the sessions.

On entering the study room, the evaluator (the second author) introduced the equipment and explained its function. The participants were given an information sheet outlining the tasks that they were about to attempt, i.e. to evacuate the building in the event of a fire alarm. In each scenario, the participants started on the ground floor of the virtual building and were requested to navigate to the staff room on the 3rd floor[32] and to wait. Three seconds after entering the staff room a fire alarm sounded. Participants were then instructed to exit the building.

Cooperative evaluation [43] was used in the evaluations. This is a concurrent think-aloud verbal protocol where users are encouraged to treat the evaluation as a shared experience with the evaluator and may ask questions at any time. As the sessions were recorded on video, the verbal protocol allowed evacuation decisions to be elicited and captured, for example any movement decisions participants made while evacuating the virtual building. Cooperative evaluation has been successfully used in a number of virtual environment evaluations [23, 39, 40].

After completing the three scenarios participants were thanked for their help and asked to complete a post-session questionnaire. This questionnaire contained questions about their satisfaction with the interface, realism of the simulation and engagement with the scenarios.

[32]In navigating to the 3rd floor of the building from the ground floor using a stairwell allowed the participants to gain experience in using the arrow keys to move in the virtual building.

$N = 12$	Scenario 1	Scenario 2	Scenario 3
Average	165.83	66	138.58
Std Dev.	82.75	31.84	70.13
Max.	350	137	328
Min.	73	34	76

Table 1: Time taken, in seconds, to evacuate the virtual building over the three scenarios.

6 Results

The main objectives of the evaluation studies were to investigate the usability and realism of the virtual environment and whether public domain game development tools are suitable for rapidly prototyping virtual environments. We were also interested in any differences in participant behaviour based on previous gaming experience.

6.1 Performance Metrics

Table 1 summarises the time taken, in seconds, to evacuate the virtual building over the three scenarios.

The scenarios were also walked through in the real building, obtaining a time of 82 seconds for scenario 1, 40 seconds for scenario 2 and 60 seconds for scenario 3. The results for the virtual environment show that the times are generally longer than in real life, which could be due to interface control issues or unfamiliarity with virtual environments, although they follow a similar pattern, i.e. scenario 2 took less time to complete than scenarios 1 and 3.

The variance in times within a scenario could be explained by the amount of previous experience the participants had with virtual environments or experience with FPS games. This could give participants an advantage when navigating in 3D virtual environments. Therefore in the pre-session questionnaire, the participants were asked to indicate their level of experience with computer games.

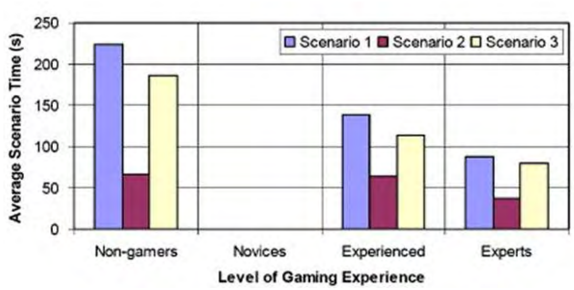

Figure 4: Average time taken in a scenario by participants of each experience level.

Figure 4 shows that the evacuation times for each scenario were affected by the amount of gaming experience of the participant, with more experienced participants completing the scenarios in less time.

However, the overall pattern for the three scenarios within an experience group stays the same.

6.2 Participant Questionnaires

Five of the questions in the post-session questionnaires provided a numerical scale for the participants to indicate preferences with 1 representing a *low* score and 7 representing a *high* score. The data gathered through these questions has been summarised in Figure 5.

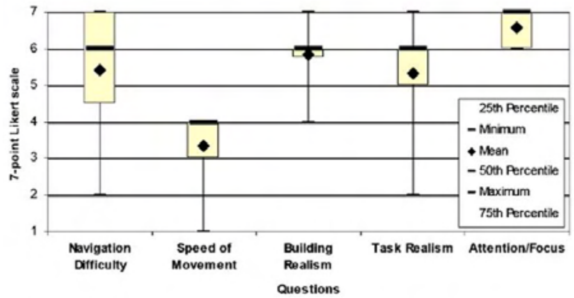

Figure 5: Box plot summary of questionnaire results.

Figure 5 shows that the responses to most of the questions did not vary greatly, as the interquartile ranges do not show a large spread. In most cases, it is the minimum values that differ most significantly from the rest of the results, i.e. there is a negative skew. The box plot for the question referring to the attention/focus of the participants shows that all participants selected 6 or 7, implying that nothing broke concentration of any participant significantly. The interquartile range for the question referring to navigation difficulty shows a larger spread of responses than for the other questions, which could be due to the amount of previous experience participants had with playing computer games or navigating virtual environments.

In addition Figure 5 shows that for the realism of the virtual environment, ratings were generally high with the mean being slightly less than six. The four participants who were the quickest to evacuate were all gamers - two experienced, two experts, three of whom rated the realism as six and the other rating it as a seven. Out of the four participants who took the longest to evacuate, there were two non-gamers and two experienced gamers. Their ratings had a slightly larger spread, one rating realism as five, two rating realism as a six and one as a seven. These results indicate that the model provided a high degree of realism. Participants who were familiar with the building being modelled, i.e. staff and students from the Computer Science department, commented that some familiar objects that they would expect to see were missing, which if present may have helped with their orientation in the virtual space.

The virtual environment was also rated on the realism of the fire evacuation task itself. The ratings were on average between five and six, but there was a larger negative skew. The four participants who were the quickest to evacuate all rated the fire evacuation task as six. However, the four participants who took the longest all gave different ratings: two, four, five and seven. Some of the comments on unrealistic elements in this area were that the fires were in unrealistic places, for example, on a stairwell. Participants commented that it was sometimes difficult to tell where the smoke was coming from and other cues such as heat indicators would have helped to make it more realistic.

6.3 Participant behaviour

Finally, the suitability of using game engines for producing training simulators was considered, with respect to participant behaviour within the fire evacuation scenarios. One possible disadvantage of using game engines for training simulators is that participants who are experienced with playing games may treat the simulation like a game, ignoring danger cues and not interacting with the environment as they would in real life.

There were two problems that did arise that may be due to the fact that the simulation had been created using a game engine. The first problem was that most participants did not expect to use a window exit in the third scenario when all the other exits were blocked. Five participants (42%) decided to look for a window to either signal for help or use it as an escape route, but the other seven participants (58%) did not try the windows until they were suggested as a possibility by the evaluator. This may have been due to most participants assuming that this kind of functionality was not available in the virtual environment, as the only other elements for interaction were doors.

The second problem was that in the third scenario, few participants avoided opening doors with smoke coming under them. This was intentionally added to the scenario to test evacuee behaviours to this dangerous situation. Nine participants (75%) either hesitated to approach because of the smoke or got so close that their avatar was injured by the heat from the fire behind the door, but still opened the door to make sure the route was blocked. Only three participants (25%) decided not to open the doors because of the smoke. In an actual fire, the heat and sound of the fire would be more apparent and would do more to discourage people from opening the doors than just the visual cues from the smoke.

In both cases, the game-based environment has interfered with the participants expectations of environment veracity. Two possible solutions are (i) a reconfiguration of input and output channels for the interaction and (ii) increasing the level of realism. For the evacuations described in this paper, user input and output was via a standard laptop. Navigation was enabled via the laptops arrow keys. As this is a common gaming key layout, changing this may avoid participants moving into a gaming mindset[33]. Also moving the output channel to a large external monitor may help users relate better to

[33] In post evaluation demonstrations of the environment, a Xbox 360 controller was used for navigation. This significantly

the first-person experience. The aim would be to modify the level of realism and immersion in the scenario to increase the degree of presence to the point where the user accepts the computer generated experience, at least temporarily, as real [21].

7 Discussion

Realism in virtual environments plays an important role in supporting any higher levels goals such as user training or accessing user awareness of, in this case hazardous, situations. In the context of fire drills, realism requirements are in (i) the built environment, (ii) the behaviour of the fire, and smoke, and (iii) the features of the scenario, for example fire placement, exit routes and crowd behaviour.

In most virtual environment developments there are tradeoffs between environment realism. In many cases this is a resource issue. Photo-realistic environments are expensive to build and computational expensive to render and display. However, particularly in this case, a non-realistic environment may influence user behaviour, for example with users treating the exercise as only a game. One way to avoid this is to import as many real environment assets into the virtual environment as possible. In the fire drill environment, floor plans were used to provide an accurate layout of the virtual building and digital photos of common textures were used to populate the virtual building, for example floor tiles, staircase features and wall posters. In this way the environment context, as a multi-storey building included expected features such as fire exit signs, thus providing a realistic context for the virtual fire drills. However it is also important that the fires encountered are also realistic.

The level of fire science used in the work presented here is limited to what was provided by the core game engine. Fire entities had associated smoke, damaged user avatars if they moved too near and over time spread within room boundaries. As evacuation times in the tested scenarios were brief, on average less than four minutes, there was little need for extensive modelling of the fires. Primarily the drills were to identify user behaviour to the presence of fire and how users navigated escape routes. If scenarios required users to be in the building longer, for example if they were trapped by fires before rescue or if the virtual building was significantly larger (i.e. [44, 46]), then there would be a requirement for more complex fire and smoke behaviour. The effects of fire and smoke on the environment, for example fire moving between floors, and the toxic effects of fumes on users would be necessary. Fortunately, game engines can facilitate some of these behaviours, for example reducing the health of users when under water in games could be modified to indicate the effects of smoke and fumes. Also many game engines provide access to source code (see [42]) so, if required, more realistic fire science behaviour could be integrated, for instance via the Fire Dynamics Simulator[34]. This would,

intensified game-like behaviour with participants regularly jumping down stairwells and running through fires.

[34]http://www.fire.nist.gov/fds/ [last access 12/11/2008].

however, be at the cost of development time and require additional developer skills.

It is also important that the definition of the scenarios themselves match user expectations. In the evaluation study, three fire drill scenarios were used. Each was designed to be unique, single participant based, with single valid escape route and having special features of interest to test participants' fire drill behaviour, for example closed doors with smoke coming underneath. Some participants felt that realism in the scenarios had been compromised by the scenario construction. For example, during the evaluations, comments were made on the placement of the fires. Participants encountered fires on opposite sides of the building, forcing them down a single escape path. In the type of building that was modelled, a fire with a number of separate start points, particularly in areas without flammable material, such as in stairways, would be very unlikely. Also the sterile nature of the virtual environment influenced the realism of the scenarios. Participants commented on *the emptiness of some corridors*, i.e. the lack of corridor clutter and other evacuees, and participants familiar with the actual building noted that some features were missing from the virtual building. Again, there is a trade-off between the accuracy of the virtual building representation, the addition of NPCs and development time. With a rapid prototyping approach some simplification is inevitable. However, the final aim of a fire drill environment will also determine building realism. If the aim is to test evacuations from a particular real world building then there will be some minimum realism requirements for the virtual representation. The virtual building must be recognisable and have features that allow training in the virtual world to influence real world behaviour. Alternatively, if the aim is to elicit participants' current fire drill behaviour or to train for behaviour improvement, then a more generic building, with set fire events and computer-based crowds, may be preferable.

There are also number of game-oriented issues that have been identified. Prior experience of interacting with virtual environments can have a notable increase in user's performance in virtual environment evaluations [6, 7]. As one of the most commonly encountered virtual environments is the modern 3D computer game, this can be problematic in environments using game technology. To balance gaming experience in such evaluations, participants could either be pre-screened for gaming ability to balance any experimental groups, or all participants could be trained [7] to an experience benchmark, for example reaching some ability threshold before commencing the actual evaluation. Both methods could be costly in requiring additional environments to be developed and in the added time to test or train participants. There are clearly cost benefit trade-offs to these issues and gathering evidence to support this is ongoing research [38].

One issue with the use of off-the-shelf technology is that of licensing. The importance of this to a development will be in the nature of the individual project. Many game engines are free to use for non-commercial applications and provide software development kits (SDKs) for this purpose (see [42]).

However, in order to access complete source code of a game engine, which may be required to integrate accurate fire science, complex NPC behaviour or custom networking, full licensing may be needed[35]. However, the nature of rapid prototyping is that the developed product is only a prototype or proof of concept, and in such cases may be thrown away before development of a full product begins. Therefore computer game technology provides an inexpensive development environment for identifying requirements and evaluating initial user behaviour to fire features and events. In some cases, resources developed for the prototype could be imported into a latter development. Although there is a lack of interchange formats between game engines, as many use proprietary formats, some game engines, such as the Source Engine, provide a text-based format. In this case map and level information could be exchanged and reused between developments.

As game engine technology develops, more realistic fire science will be available within core game engines. Also complex NPC behaviour is now becoming available for addition into game modifications. Finally, there is an expanding game modification community worldwide, producing ever more complex game modifications and third-party tools to integrate new and existing technology into game engines. Linking realistic fire science to the usability and robustness of game technology has great promise for rapidly prototyping simulations of custom buildings for fire evaluation drills.

8 Conclusions and Future Work

Virtual environments can support the training and observation of fire evacuee behaviours in virtual 3D models based on real world buildings. However complex virtual environments can be difficult to build. This paper has shown how the reuse of computer game technology can aid the rapid prototyping of a fire evacuation virtual environment. Over a three week period, a single developer constructed a realistic model of a real world building which was populated with fire drill evacuation scenarios and evaluated. While participants found the simulated environment realistic, performance metrics indicated clustering in the results based on participants gaming experience.

It was found that a participants previous experience at playing computer games played an important part in their experience of a game engine-based simulation. Therefore, this could affect results in other virtual environment simulations.

Some dangerous behaviour was exhibited by participants, for example ignoring smoke coming from under doors. This is problematic if good fire evacuation procedures are to be learnt. This may require virtual environments to have a variety of visual or audio cues to compensate the lack of heat and odour

[35] Actual costs for full licensing are difficult to determine as game developers commonly require a non-disclosure agreement as part of the negotiation. One exception is the licensing for Jupiter Ex with licensing packages ranging from $US\$15,000$ to $US\$55,000$ depending on developer options.

cues found in real environments, for instance an indication of doors becoming hot with a red handle.

Future work includes investigating features in the environment that will increase the realism of the experience. One limitation of this virtual environment was that the fire extinguishers could not be used. Four participants wanted to try and use the fire extinguishers, but this feature was not implemented. Half-Life2 does not provide a functional fire extinguisher that can put out fires. Although the use of fire extinguishers may be secondary to evacuating a building, we are investigating adding this functionality by altering the source code of the Source Engine.

Finally when people are attempting to exit a building, an individuals actions may affect other peoples actions. For example, an individuals choice of exit may not just depend on how close an exit is; it may also depend on how many people are heading towards an exit, e.g. a crowd may make an exit difficult to reach [20]. The Source Engine provides the functionality for limited artificial intelligent NPCs which can be instructed to navigate through the environment based on event triggers. We intend to explore the possibility of adding NPCs into the evacuation scenarios and exploiting the multi-player networked facilities of Half-Life2 to enable multiple human participants to experience the same fire drill scenario concurrently.

9 Acknowledgements

The authors would like to thank the evaluation study participants, Dr Liz Burd (Computer Science, Durham University), the HEFCE Centre for Excellence in Teaching and Learning in Computing (Active Learning in Computing, Durham University) and Brendan Hodgson (Computer Science, Durham University). The authors are also grateful to the reviewers who provided valuable comments for the improvement of this paper. This work was funded in part by the Nuffield Foundation (grant URB/34118).

References

[1] Jan M. Allbeck and Norman I. Badler. Embodied autonomous agents. In K. M. Stanney, editor, *Handbook of Virtual Environments*, pages 313–332. Lawrence Erlbaum Associates, New Jersey, 2002.

[2] Eric Badiquè, Marc Cavazza, Gudrun Klinker, Gordon Mair, Tony Sweeney, Daniel Thalmann, and Nadia M. Thalmann. Entertainment applications of virtual environments. In Kay M. Stanney, editor, *Handbook of Virtual Environments*, pages 1143–1166. Lawrence Erlbaum Associates, New Jersey, 2002.

[3] John T. Bell and H. Scott Fogler. Implementing virtual reality laboratory accidents using the Half-Life game engine, WorldUp, and Java3D. In *2003 American Society for Engineering Education Annual Conference & Exposition*. American Society for Engineering Education, 2003.

[4] Kristopher J. Blom and Steffi Beckhaus. Supporting the creation of dynamic, interactive virtual environments. In Stephen N. Spencer, editor, *ACM Symposium on Virtual Reality Software and Technology (VRST 2007)*, pages 51–54, California, USA, 2007. ACM.

[5] Doug A. Bowman, Ernst Kruijff, Joseph J. LaViola Jr., and Ivan Poupyrev. *3D User interfaces: Theory and Practise*. Addison Wesley, USA, 2005.

[6] L. Enochsson, B. Isaksson, R. Tour, A. Kjellin, L. Hedman, T. Wredmark, and L. Tsai-Felländer. Visuospatial skills and computer game experience influence the performance of virtual endoscopy. *Journal of Gastrointestinal Surgery*, 8(7):874–880, 2004.

[7] Andreas Frey, Johannes Hartig, André Ketzel, Axel Zinkernagel, and Helfried Moosbrugger. The use of virtual environments based on a modification of the computer game Quake III Arena in psychological experimenting. *Computers in Human Behaviour*, 23:2026–2039, 2007.

[8] E. R. Galea and J. M. Perez Galparsoro. A computer-based simulation model for the prediction of evacuation from mass-transport vehicles. *Fire Safety Journal*, 22:341–366, 1994.

[9] Erik Guilfoyle. *Half-Life 2 Mods for Dummies*. Wiley Publishing, Inc, 2007.

[10] S. Gwynne, E. R. Galea, P. J. Lawrence, and L. Filippidis. Modelling occupant interaction with fire conditions using the buildingEXODUS evacuation model. *Fire Safety Journal*, 36:327–357, 2001.

[11] S. Gwynne, E. R. Galea, M. Owen, P. J. Lawrence, and L. Filippidis. A review of the methodologies used in the computer simulation of evacuation from the built environment. *Building and Environment*, 34:741–749, 1999.

[12] C. W. Johnson. Using evacuation simulations to ensure the safety and security of the 2012 Olympic venues. *Safety Science*, 46(2):302–322, 2008.

[13] C. W. Johnson and L. Nilsen-Nygaard. Extending the use of evacuation simulators to support counter-terrorism: Using models of human behaviour to coordinate emergency responses to improvised explosive devices. In R. J. Simmons, D. J. Mohan, and M. Mullane, editors, *Proceedings of the 26th International Conference on Systems Safety, Vancouver, Canada 2008*, Unionville, VA, USA, 2008. International Systems Safety Society.

[14] G. Drew Kessler. Virtual environment models. In K. M. Stanney, editor, *Handbook of Virtual Environments*, pages 255–276. Lawrence Erlbaum Associates, New Jersey, 2002.

[15] S. Ko, M. Spearpoint, and A. Teo. Trial evacuation of an industrial premises and evacuation model comparison. *Fire Safety Journal*, 42:91–105, 2007.

[16] Blazej Kot, Burkhard Wuensche, John Grundy, and John Hosking. Information visualisation utilising 3D computer game engines case study: A source code comprehension tool. In *6th ACM SIGCHI New Zealand chapter international conference on computer-human interaction (CHINZ 2005)*, pages 53–60. ACM, 2005.

[17] John E. Laird. Research in human-level AI using computer games. *Communications of the ACM*, 45(1):32–35, 2002.

[18] George Lepouras and Costas Vassilakis. Virtual museums for all: employing game technology for edutainment. *Virtual Reality*, 8:96–106, 2005.

[19] Michael Lewis and Jeffrey Jacobson. Game engines in scientific research. *Communications of the ACM*, 45(1):27–31, January 2002.

[20] S. M. Lo, H. C. Huang, P. Wang, and K. K Yuen. A game theory based exit selection model for evacuation. *Fire Safety Journal*, 41:364–369, 2006.

[21] Michael N. Louka and Claudio Balducelli. Virtual reality tools for emergency operation support and training. In *Proceedings of TIEMS (The International Emergency Management Society)*, Oslo, June 2001.

[22] B. Mac Namee, P. Rooney, P. Lindstrom, A. Ritchie, F. Boylan, and G. Burke. Serious Gordon: Using serious games to teach food safety in the kitchen. In *9th International Conference on Computer Games: AI, Animation, Mobile, Educational & Serious Games CGAMES06*, 2006.

[23] Tim Marsh and Peter Wright. Co-operative evaluation of a desktop virtual reality system. In S. P. Smith and M. D. Harrison, editors, *Workshop on User Centered Design and Implementation of Virtual Environments*, pages 99–108. The University of York, 1999.

[24] Thiago Tinoco Pires. An approach for modeling human cognitive behavior in evacuation models. *Fire Safety Journal*, 40:177–189, 2005.

[25] George V. Popescu, Grigore C. Burdea, and Helmuth Trefftz. Multimodal interaction modelling. In K. M. Stanney, editor, *Handbook of Virtual Environments*, pages 435–454. Lawrence Erlbaum Associates, New Jersey, 2002.

[26] Salvador Casadesús Pursals and Federico Garriga Garzón. Optimal building evacuation time considering evacuation routes. *European Journal of Operational Research*, 192:692–699, 2009.

[27] D. A. Purser and M. Bensilum. Quantification of behaviour for engineering design standards and escape time calculations. *Safety Science*, 38:157–182, 2001.

[28] T. X. Qin, Y. C. Guo, C. K. Chan, and W. Y. Lin. Numerical simulation of the spread of smoke in an atrium under fire scenario. *Building and Environment*, 44:56–65, 2009.

[29] Aizhu Ren, Chi Chen, and Yuan Luo. Simulation of emergency evacuation in virtual reality. *Tsinghua Science and Technology*, 13(5):674–680, 2008.

[30] Aizhu Ren, Chi Chen, Jianyong Shi, and Liang Zou. Application of virtual reality technology to evacuation simulation in fire disaster. In Hamid R. Arabnia, editor, *Proceedings of the 2006 International Conference on Computer Graphics and Virtual Reality*, pages 15–21, USA, 2006. CSREA Press.

[31] Geneviève Robillard, Stéphane Bouchard, Thomas Fournier, and Patrice Renaud. Anxiety and presence using VR immersion: A comparative study of the reactions of phobic and non-phobic participants in therapeutic virtual environments derived from computer games. *CyberPsychology & Behaviour*, 6(5):467–475, 2003.

[32] Tzu-Sheng Shen. ESM: a building evacuation simulation model. *Building and Environment*, 40:671–680, 2005.

[33] Tzu-Sheng Shen, Yu-Hsiang Huang, and Shen-Wen Chien. Using fire dynamic simulation (FDS) to reconstruct an arson fire scene. *Building and Environment*, 43:1036–1045, 2008.

[34] Naai-Jung Shih, Ching-Yuan Lin, and Chih-Hsiang Yang. A virtual-reality-based feasibility study of evacuation time compared to the traditional calculation method. *Fire Safety Journal*, 34:377–391, 2000.

[35] Mohd.Fairuz Shiratuddin and Walid Thabet. Virtual office walkthrough using a 3D game engine. *International Journal of Design Computing*, 4, 2002.

[36] Mel Slater, Vasilis Linakis, Martin Usoh, and Rob Kooper. Immersion, presence and performance in virtual environments: An experiment with tri-dimensional chess. *ACM Virtual Reality Software and Technology (VRST'96)*, pages 163–172, 1996.

[37] Shamus P. Smith and David J. Duke. Binding virtual environments to toolkit capabilities. *Computer Graphics Forum*, 19(3):C–81–C–89, 2000.

[38] Shamus P. Smith and Sam Du'Mont. Measuring the effect of gaming experience on virtual environment navigation tasks. 2008. Submitted.

[39] Shamus P. Smith and Jonathan Hart. Evaluating distributed cognitive resources for wayfinding in a desktop virtual environment. In Y. Kitamura, D. Bowman, B. Fröhlich, and W. Stürzlinger, editors, *IEEE Symposium on 3D User Interfaces 2006*, pages 3–10. IEEE, 2006.

[40] Shamus P. Smith and Susan Todd. Usability evaluation of a haptic-based clinical skills training system. *International Journal of Clinical Skills*, pages 39–44, 2008.

[41] Chieh-Hsin Tang, Wu-Tai Wu, and Ching-Yuan Lin. Using virtual reality to determine how emergency signs facilitate way-finding. *Applied Ergonomics*, 2008. doi:10.1016/j.apergo.2008.06.009.

[42] David Trenholme and Shamus P. Smith. Computer game engines for developing first-person virtual environments. *Virtual Reality*, 12:181–187, 2008.

[43] Peter C. Wright and Andrew F. Monk. A cost-effective evaluation method for use by designers. *International Journal of Man-Machine Studies*, 35(6):891–912, 1991. Academic Press Ltd.

[44] J. P. Yuan, Z. Fang, Y. C. Wang, S. M. Lo, and P. Wang. Integrated network approach of evacuation simulation for large complex buildings. *Fire Safety Journal*, 2008. doi:10.1016/j.firesaf.2008.07.004.

[45] Weifeng Yuan and Kang Hai Tan. An evacuation model using cellular automata. *Physica A*, 384(2):549–566, 2007.

[46] Maohua Zhong, Congling Shi, Xuwei Tu, Tairan Fu, and Li He. Study of the human evacuation simulation of metro fire safety analysis in China. *Journal of Loss Prevention in the Process Industries*, 21:287–298, 2008.

[47] Michael Zyda. From visual simulation to virtual reality to games. *IEEE Computer*, pages 25–32, September 2005.

10 Figure Captions

Figure 1: Virtual model view (left) including fire entities and accompanying smoke and real world view (right) of a stairwell.

Figure 2: The completed model geometry in the Hammer editor.

Figure 3: Virtual model (left) and real building (right) views of the Computer Science department.

Figure 4: Average time taken in a scenario by participants of each experience level.

Figure 5: Box plot summary of questionnaire results.